Automation in Construction, Vol. 13, No. 4, 2004, pp. 481-490

A two-stage dynamic model on allocation of construction facilities with genetic algorithm

K.W. Chau

Department of Civil & Structural Engineering, Hong Kong Polytechnic University, Hunghom, Kowloon, Hong Kong

Abstract

By their very nature, activities within the construction site are generally highly dynamic and complex. Hence it is highly desirable to be able to formulate the optimal strategy for allocating site-level facilities at different times of the project. The principal objective is to minimize the total cost, which comprises the transportation, handling, capital, and operating costs at potential intermediate transfer centers of various plant and material resources over the entire project duration. The problem can be formulated as a mixed-integer program, which entails enormous computational effort for the solution, in particular when the problem size is large. In this paper, a two-stage dynamic model is developed to assist construction planners to formulate the optimal strategy for establishing potential intermediate transfer centers for site-level facilities such as batch plants, lay-down yards, receiving warehouses, various workshops, etc. Under this approach, the solution of the problem is split into two stages, namely, a lower-level stage and an upper-level stage. The former can be solved by a standard linear programming method whereas the latter is solved by a genetic algorithm. The efficiency of the proposed algorithm is demonstrated through case examples.

Keywords: Construction facilities, Dynamic resources allocation, Genetic Algorithm, Mixed-integer program, Site planning, Two-stage formulation

Introduction

It may not be efficient to distribute resources directly from outside sources to their final locations in a construction site. Intermediate transfer centers can be established in order to lower transportation and handling costs. The demands on these construction facilities may change over the project duration whilst the transportation and handling costs may vary at different locations as well as over time period. During the past two decades, various investigators have extensively studied this facility location problem [1-3]. Yet, in order to simplify the problem, they assume the situation to be static in which the costs for establishment and operation of transfer centers and for transportation do not change with time. Their assumptions may be justified under the technology level and nature of projects at that time.

Nowadays, construction projects are usually becoming larger in size, involving more interdisciplinary fields and increasing in complexity. At the same time, the recent advancement in computer technology furnishes the availability of efficient numerical models on microcomputers. As such, there is a pressing necessity to develop a robust dynamic model in order to aid in decision making for optimal strategy in allocating the construction resources in relation to the time dimension. Drezner and Wesolowsky [4] proposed a solution algorithm for the facility location problem among a given set of demand points with time-dependent weights associated with each demand point. Hinojosa et al. [5] employed a lagrangian relaxation algorithm to solve the multi-period, multi-commodity facility location problem.

Canel et al. [6] suggested a solution algorithm integrating the branch-and-bound technique with dynamic programming in solving the facility location problem. In the construction field, studies have also been made on planning of temporary facilities. Cheng and OConnor [7] developed a software named ArcSite with enhanced GIS for construction site layout. Zouein and Tommelein [8] performed dynamic layout planning using a hybrid incremental solution method. Li and Love [9] employed a genetic search for solving construction site-level unequal-area facility layout problems. Chau et al. [10] implemented visualization as planning and scheduling tool in construction. Chau et al. [11] applied data warehouse and decision support system in construction management.

During the last few years, artificial intelligence (AI) techniques have also been incorporated to solve the dynamic problems. Son and Skibniewski [12] addressed the determination of optimal location of fixed base construction equipment (e.g. cranes) on the construction site with the use of genetic algorithms. Elwany et al. [13] studied the coupled integration of AI and optimization techniques in the facility layout problem. Antunes and Peeters [14] employed the simulated annealing method in the dynamic location problems. Elbeltagi and Hegazy [15] developed a hybrid AI-based system for site layout planning in construction. Chau and Anson [16] implemented a knowledge-based system for construction site level facilities layout. The genetic algorithms, by mimicking the mechanisms of biological genetics and natural selection [17], have been applied in many domain problems in recent years [18-21].

Problem Statement

In this paper, a two-stage dynamic model is developed to assist construction planners to formulate the optimal strategy for establishing potential intermediate temporary site-level facilities that support construction, such as batch plants, lay-down yards, receiving warehouses, various workshops, etc. The principal objective is to identify which if any combinations of a pre-enumerated set of construction site-level transfer facilities are worth establishing for any time interval during the course of a project. The worth of setting up of a specific temporary facility is assessed by means of an objective function that minimizes the total cost, which comprises the transportation, handling, capital, and operating cost at potential intermediate transfer centers of various plant and material resources over the entire project duration. Under this approach, the solution of the problem is split into two stages, namely, a lower-level stage and an upper-level stage. The former can be solved by a standard linear programming method whereas the latter is solved by a genetic algorithm. Through benchmark comparison with the solution by the conventional mixed-integer program, the efficiency of the proposed algorithm is demonstrated with case examples.

Formulation of Problem

The problem can be formulated as a mixed-integer program [22], which entails enormous computational effort for the solution, in particular when the problem size is large. The problem of mixed-integer program is written as:

$$\text{Minimize C} = \sum_{t=1}^{T}\sum_{f=1}^{F}\sum_{j=1}^{J}\sum_{i=1}^{I}\alpha_t c_{ijft}^{u} u_{ijft} + \sum_{t=1}^{T}\sum_{f=1}^{F}\sum_{k=1}^{K}\sum_{j=1}^{J}\alpha_t c_{jkft}^{v} v_{jkft} + \sum_{t=1}^{T}\sum_{f=1}^{F}\sum_{k=1}^{K}\sum_{i=1}^{I}\alpha_t c_{ikft}^{w} w_{ikft}$$

$$+\sum_{t=1}^{T}\sum_{j=1}^{J}\alpha_t c_{jt}^{x} x_{jt} + \sum_{t=1}^{T}\sum_{j=1}^{J}\alpha_t c_{jt}^{y} y_{jt} + \sum_{t=1}^{T}\sum_{j=1}^{J}\alpha_t c_{jt}^{z} z_{jt} + \sum_{t=1}^{T}\sum_{f=1}^{F}\sum_{j=1}^{J}\alpha_t c_{jft}^{q} q_{jft} \quad (1)$$

subject to the following constraints:

$$S_{ift} = \sum_{j=1}^{J} u_{ijft} + \sum_{k=1}^{K} w_{ikft}\text{, for i = 1,2,…,I; f = 1,2,…,F; t = 1,2,…,T} \tag{2}$$

$$D_{kft} = \sum_{j=1}^{J} v_{jkft} + \sum_{i=1}^{I} w_{ikft}\text{, for k = 1,2,…,K; f = 1,2,…,F; t = 1,2,…,T} \tag{3}$$

$$q_{jft} = \sum_{i=1}^{I} u_{ijft} = \sum_{k=1}^{K} v_{jkft} \le Az_{jt}\text{, for j = 1,2,…,J; f = 1,2,…,F; t = 1,2,…,T} \tag{4}$$

$$z_{jt} - z_{j,t-1} + y_{jt} - x_{jt} = 0\text{, for j = 1,2,…,J; t = 1,2,…,T} \tag{5}$$

$$x_{jt}, y_{jt}, z_{jt} \in (0,1) \tag{6}$$

$$u_{ijft}, v_{jkft}, w_{ikft} \ge 0 \tag{7}$$

$$\alpha_t = \frac{1}{(1+\alpha)^{(t-1)}} \tag{8}$$

where I is the number of sources; J is the number of potential transfer centers; K is the number of destinations; F is the total number of different types of site-level facilities each with a type number f (i.e., batch plant, lay-down yard, receiving warehouse, various workshops, and so on, with their definitions shown as a typical constructed-oriented example in Table 1); T is the number of periods of the problem; α_t is the discount rate for period t; c^u_{ijft} , c^v_{jkft} , and c^w_{ikft} are the costs of transportation from source i to transfer center j, from transfer center j to destination k, and directly from source i to destination k during period t for facility type f; u_{ijft} , v_{jkft} , and w_{ikft} are the quantities of construction resources (materials or plant) that are transported from source i to transfer center j, from transfer center j to destination k, and directly from source i to destination k for site-level facility of type f during period t; c^x_{jt} , c^y_{jt} , and c^z_{jt} are the startup cost, closure cost, and fixed operation cost at transfer center j during period t; z_{jt} denotes the existence of transfer center j during period t (z_{jt} = 1 if the transfer center is in operation and 0 otherwise, with the initial condition of z_{j0} = 0); x_{jt} denotes the opening of transfer center j during period t (x_{jt} = 1 if $z_{jt} - z_{j,t-1} = 1$ and 0 otherwise); y_{jt} denotes the closure of transfer center j during period t (y_{jt} = 1 if $z_{jt} - z_{j,t-1} = -1$ and 0 otherwise); c^q_{jft} is the variable cost of operation at transfer center j during time period t for facility type f; q_{jft} is the quantity of resources for facility type f at transfer center j during period t; S_{ift} and D_{kft} are the quantities of resources for facilities of type f, as generated at source i and in demand at destination k during period t; A is an arbitrary constant with very large value (of order larger than q_{jft}); α is an appropriate discount rate to determine the net present value.

In equation (1), the objective function represents the total operation cost of the problem. The first three terms denote the transportation costs, the subsequent four terms account for the opening costs, closing costs, fixed costs, and variable operative costs, respectively of the transfer centers. Equations (2) and (3) represent constraints on conservation for movement of resources through site-level facilities. Equation (4) denotes constraints on the total quantity of resources at a potential transfer center: if z_{jt} = 0, no resource for facility f is allowed to be transported via this center, i.e. q_{jft} = 0; and, if z_{jt} = 1, the total quantity of resource deliveries into and out of the transfer center is then computed. Equation (5) represents the relationships between opening, operation, and closure of a transfer center which satisfies the conditions shown in Table 2. Equation (6) specifies the binary nature of the parameters whilst equation (7) ensures that flows in the network are always positive. Equation (8) denotes the discount equation to determine the net present value.

Since it is too complicated to incorporate a more complex handling function, an assumption is made here that the cost for handling is a linear function of quantity for this model. By considering the problem to be one of allocation, each transfer station is modeled at a point location and its surface area and shape are ignored. It has been considered in the model that transfer facility may not use up its maximum capacity at all times and there will be variations at different times.

Split-Step Algorithm

In this approach, the conventional mixed-integer programming is split into two steps, namely, a lower-level step and an upper-level step, which is still capable to furnish a global optimal solution. The time-dependent dynamic allocation is dealt with in the upper-level step whilst the lower-level step focuses on the instantaneous transportation problem. The advantage of this approach is that it can reduce substantially the requisite computational efforts.

Lower-level Step

The lower-level step is formulated as a static transportation problem for a specific instant. For each period t, the operation status of variable z_{jt}, representing the set of transfer centers that are in operation for that particular time, is acquired from the upper-level step. A network connecting the sources, transfer centers, and destinations is established. Optimization of this network can be solved easily by the standard linear programming solution in many commercially available packages. The shortest path for each pair of source and destination can be determined by employing any standard shortest path algorithm. When a transfer center j is involved, the variable cost of operating the center c^{q}_{jft} for facility type f is added to the path. The resources for facility type f will be transported via that transfer center if the sum of the transport cost and variable operating cost is the smallest for that path.

For each period t and facility type f, the problem to minimize the total transport cost plus variable operation cost C_{ft} is written as:

$$\underset{m_{ft}}{Minimize}\, C_{ft} = \sum_{i=1}^{I}\sum_{k=1}^{K} c^{\min}_{ikft} m_{ikft} \tag{9}$$

subject to the following constraints:

$$\sum_{k=1}^{K} m_{ikft} = S_{ift}\text{, for i = 1,2,…,I} \tag{10}$$

$$\sum_{i=1}^{I} m_{ikft} = D_{kft}\text{, for k = 1,2,…,K} \tag{11}$$

$$m_{ikft} \geq 0\text{, for i = 1,2,…,I; k = 1,2,…,K} \tag{12}$$

where m_{ft} is a collection of the movement variable m_{ikft} representing the quantity of resources delivered from source i to destination k for facility type f in period t and $c_{jkft}^{\min}$ is the minimum unit transport cost plus the variable operation cost at the transfer center skimmed from the network for each facility type f from each source i to each destination k in period t. This linear programming problem can be solved by any standard routine.

The distribution of resource delivery is then solved by the transportation problem. The quantity of resource deliveries at transfer center j is then determined using the delta function and the resource delivery for that facility from each source i to destination k:

$$q_{jft} = \sum_{i=1}^{I}\sum_{k=1}^{K} \delta_{ijkft} m_{ikft}\text{, for j = 1,2,…,J} \tag{13}$$

where the delta function δ_{ijkft} defines whether or not the shortest path of resource delivery for facility type f from source i to destination k will pass via center j. It has a value of 1 if the path is via the transfer center j and 0 otherwise.

In cases total cost for using a transfer station (comprising transportation, opening, operation, and closing costs) is more than transportation cost for direct transfer from the source to the destination, the delta function δ_{ijkft} for that transfer station will take a value of 0 representing that it is less competitive. In other words, the delta function δ_{ijkft} will only take a value of 1 for the shortest path from the source to the destination.

In some situation, the use of a transfer station may also be related to the business process (for example, a receiving function) and this demand must be satisfied by using at least one transfer station. This requirement can be satisfied by adding a constraint that the delta function $\sum_{j=1}^{J} \delta_{ijkft} = 1$, for i = 1,2,…,I; k = 1,2,…,K. This will force the set up of at least one transfer station even when the transportation cost for direct transfer option is the lowest.

Upper-level Step

The output of the lower-level step (i.e., the transportation cost and variable operating costs for each individual period) will in turn become the input in the upper-level step. The purpose of this step is to optimize the total cost incurred by the system. In addition to the transportation cost and variable operating costs acquired from the lower-level step, it also comprises costs of establishment, closedown, and fixed operation costs for different combinations of transfer centers in different periods. The step is formulated as:

$$\underset{z}{Minimize}\ \ C = \sum_{t=1}^{T} \alpha_t \left\{ \sum_{j=1}^{J} \left[c_{jt}^{x} x_{jt}(z) + c_{jt}^{y} y_{jt}(z) + c_{jt}^{z} z_{jt} \right] + \sum_{f=1}^{F} C_{ft}(z) \right\} \tag{14}$$

where $z = \{ z_{jt}$, j=1,2,…,J; t = 1,2,…,T} is a collection of the existence of transfer center j during period t; $C_{ft}(z)$ is the transportation and variable operation costs as determined by the lower-level step based on a particular **z**, and $x_{jt}(z)$, and $y_{jt}(z)$ are the opening and closure of a transfer center j in period t, which can be obtained from **z** and have values of either 1 or 0. Thus, the time-dependent existence of transfer center z_{jt} , representing whether or not a transfer center is operating at that moment, is employed as the sole decision variable in this step.

As such, the upper-level step is an unconstrained minimization process with binary variables only. Exhaustive enumeration method may be used for a small-sized problem, but not for a medium-sized problem. For a case with a period of study of 5 years and 5 transfer center locations, the number of combinations is $2^{(5*5)}$ which equals 3.3 x 10^7. Genetic algorithm, which has been proven to be computationally very efficient in arriving at the global solution, is employed in this step to solve the optimization problem. The process comprises initialization, reproduction, crossover, mutation and iteration.

The population is first initialized with the number of generation n set equal to 1. The other parameters, including the length of a solution string M_S (= J.T), the population size for the problem M_P, pool size for the reproduction process M_R, and mutation rate μ, are then set. The population of trial solution strings is then generated randomly $\Omega_p^n = (\varphi_{ps}^n$, s = 1,2,…, M_S), p = 1,2,…, M_P, with each $\varphi_{ps}^n \in (0,1)$ corresponding to a particular z_{jt} variable in the upper-level step. The objective value in equation (14) is then evaluated for each trial solution $C_p^n = C(\Omega_p^n)$, p = 1,2,…, M_P through solving the lower-level step.

Amongst the trial solutions, M_R of them are then randomly selected and their objective values are compared. The best solution from this pool is selected and reproduced in the (n+1)th generation. This reproduction process is repeated until M_P trial solutions are acquired in this new generation (Ω_p^{n+1} , p = 1,2,…, M_P).

Uniform crossover is employed here and a binary crossover mask (m_s, s = 1,2,…, M_S) is generated randomly. Two trial solutions are randomly selected from the population, $\Omega_{p'}^{n+1}$ and $\Omega_{p''}^{n+1}$. If m_s = 0, no change occurs; however, if m_s = 1, the values of $\varphi_{p's}^{n+1}$ and $\varphi_{p''s}^{n+1}$ are exchanged. This crossover process is continued until exhaustion of the entire population.

Amongst the population of trial solution, $m_m = \mu M_P M_S$ bits are randomly selected for mutation. If the mutation positions are (p_m, s_m), m = 1,2,…, m_m, the value of each $\varphi_{p_m s_m}^{n+1}$ is reverted from 0 to 1 or vice versa. The objective value in equation (14) is again evaluated for each trial solution $C_p^{n+1} = C(\Omega_p^{n+1})$, p = 1,2,…, M_P. The optimal solution for this generation is z* = Ω_{p*}^{n+1} , where p* is the solution corresponding to the minimum objective value.

The iteration to reproduction for the next generation continues until either one of the termination criteria is met, i.e., maximum tolerance between maximum and minimum values β, or maximum number of iterations N. If C_{max} = Max(C_p^{n+1} , p =1,2,…, M_P) and C_{min} = Min(C_p^{n+1} , p =1,2,…, M_P) and $|C_{max} - C_{min}| < \beta$ or if (n+1) > N, the optimal solution at (n+1)th generation becomes the solution to the problem.

Model input and outputs

The model is not biased in itself towards whether or not handling at the point of use is more efficient. The user is required to enter the numbers of sources, potential transfer centers, destinations, facilities, and time periods, as well as the costs of transportation, operating, setting up, and closure of any potential transfer centers. Then the model can output the optimized arrangement of these transfer centers together with their opening and closing times during the project duration.

Practical considerations

This model can assist the user to evaluate different possible locations of transfer centers through scenario analysis. After having identified feasible locations, the user has to input unit total costs for that particular location to the model. The merits of access to the point of demand (for example, entry into a building) can be reflected in terms of unit costs. A reduction in unit cost can be made for this specific location or different amount of penalty unit costs can be imposed on other less convenient locations. The space availability of many construction sites can be considered by adding constraint conditions to define the maximum capacity of each transfer station in the objective functions. Different amount of penalty unit costs can also be imposed on various layout preferences.

Numerical Examples

A simple hypothetical case with only a single type of site-level facility (concrete delivery in this case) is considered first in order to demonstrate the effectiveness of the proposed algorithm. More types of site-level facility as shown in Table 1 can also be incorporated in a similar manner for a more complicated situation. In a construction site with a project period of three years, there are three sources of raw materials (S1, S2, and S3), three potential transfer centers representing batch plant locations (T1, T2 and T3), and four destinations of placement (D1, D2, D3 and D4) for concrete delivery (F1). The objective is to determine the most optimized solution by ordering concrete from its raw sources directly or establishing any combinations of up to three concrete batch plants on site and also the temporal distributions during these three years period. Figure 1 shows the schematic layout of the worksite for this example. Table 3 shows the quantities of supply and demand of concrete at various sources and destinations, respectively whilst Table 4 displays the opening, closure, fixed, and variable operation costs for the concrete batch plants. Table 5 lists the unit transportation costs amongst sources, concrete batch plants and destination. Batch plant locations T1, T2, and T3 have maximum annual capacities of 2500, 1200 and 1200 cubic meters, respectively. The prevailing discount rate is 7% per year. The optimal strategies for operation of these transfer centers are then determined by this split-step algorithm, with values verified by employing mixed-integer programming.

The parameters for the genetic algorithm are selected in the light of experience gleaned from various literatures [16-20] supplemented by some experimental trial and errors. The population and pool sizes are chosen to provide sufficient sampling of the decision space yet to limit the computational burden simultaneously. A mutation rate is applied so as to avoid being trapped in local optima and to preserve the diversification among the population in the search, but not to be excessive that may lead to large fluctuation or non-convergence. In this study, the population size M_P selected is 25, the pool size for reproduction process M_R used is 6, and the mutation rate μ adopted is 1%. These values are consistent with other empirical studies with high crossover probability, low mutation probability, and moderate population size, although it is found from experimental trial and errors that this genetic algorithm is not highly sensitive to these parameters.

The chromosomes are represented by binary strings of length with 9 bits. The first three, the second three, and the last three digits denote the operational state for T1, T2, and T3, respectively. The optimal solution shown as a chromosome from the genetic algorithm is

"111011001"

representing that the optimal strategy is to open transfer centers T1, T2, and T3 in the first year, second year, and third year, respectively until the end of the project. Table 6 shows the detailed solution on concrete deliveries amongst sources, transfer centers, and destinations for this example. Figure 2 shows the solution by the split-step algorithm versus number of generation. It is found that the split-step algorithm is capable to accomplish exactly the same solution (total cost of $39.08 million) as that by the mixed-integer programming.

In order to demonstrate the computational efficiency of the split-step algorithm, several cases with a diversity of problem sizes as shown in Table 7 are evaluated by comparison with the mixed-integer programming as a benchmark. The computing processing unit time, representing the actual running time executed by a Pentium IV 1.7G personal computer, is used to compare the computational efficiency between the mixed integer programming and the proposed model. Figure 3 shows the relationships between the computing processing unit time versus the multiplication of the period of study times the number of transfer center locations for both the mixed-integer programming and the split-step algorithm. As shown in Figure 3, one of the advantages of this split-step algorithm is its small rate of increase of the computing processing unit time even for a large number of transfer centers. It can be observed that the relative advantage of the split-step algorithm becomes more acute when the problem size becomes larger.

Conclusions

A two-stage dynamic optimization model has been developed which is able to assist in the formulation of the optimal strategy on allocating potential transfer centers for resources via temporary site-level facilities on construction site over the project period. In the algorithm, the lower-level step representing an instantaneous time is solved by standard linear programming whilst the dynamic existences of transfer center z_{jt} representing the operation status of the transfer centers are dealt with in the upper-level step by genetic algorithm. It has been shown that, for a medium size problem, this algorithm is robust in locating a global optimum from the solution space, yet entailing a much less amount of computational effort than that by employing the conventional mixed-integer program.

Acknowledgements

This research was supported by the Research Grants Council of Hong Kong (PolyU5060/99E).

References

[1] J. Barcelo, J. Casanovas, A heuristic lagrangian algorithm for the capacitated plant location problem, European Journal of Operational Research 15 (2) (1984) 212-226.
[2] J.G. Klincewicz, H. Luss, A Lagrangian relaxation heuristic for capacitated facility location with single-source constraints, Journal of Operational Research Society 37 (5) (1986) 495-500.
[3] J. Krarup, P.W. Pruzan, The simple plant location problem: survey and synthesis, European Journal of Operational Research 12 (1) (1983) 36-81.
[4] Z. Drezner, G.O.Wesolowsky, Facility location when demand is time dependent, Naval Research Logistics 38 (5) (1991) 763-777.
[5] Y. Hinojosa, J. Puerto, F.R. Fernandez, A multiperiod two-echelon multicommodity capacitated plant location problem, European Journal of Operational Research 123 (2) (2000) 271-291.
[6] C.Canel, B.M. Khumawala, J. Law, A. Loh, An algorithm for the capacitated, multi-commodity multi-period facility location problem, Computers and Operations Research 28 (5) (2001) 411-427.
[7] M.Y. Cheng, J.T. OConnor, ArcSite: enhanced GIS for construction site layout, Journal of Construction Engineering and Management, ASCE, 122 (4) (1996) 329-336.
[8] P.P. Zouein, I.D. Tommelein, Dynamic layout planning using a hybrid incremental solution method, Journal of Construction Engineering and Management, ASCE 125 (6) (1999) 400-408.
[9] H. Li, P.E.D. Love, Genetic search for solving construction site-level unequal-area facility layout problems, Automation in Construction 9 (2) (2000) 217-226.
[10] K.W. Chau, M. Anson, J.P. Zhang, Implementation of visualization as planning and scheduling tool in construction, Building and Environment 38 (5) (2003) 713-719.
[11] K.W. Chau, Y. Cao, M. Anson, J.P. Zhang, Application of data warehouse and decision support system in construction management, Automation in Construction 12 (2) (2003) 213-224.
[12] J. Son, M. Skibniewski, Genetic algorithms in construction engineering and management computations, International Journal of Construction Information Technology 3 (2) (1995) 1-27.
[13] H. Elwany, M.G. Abou-Ali, N.A. Harraz, The layout problem: investigation and aggregation of artificial intelligence and optimization techniques, Lecture Notes in Artificial Intelligence, Springer-Verlag 1531 (1998) 530-541.
[14] A. Antunes, D. Peeters, On solving complex multi-period location models using simulated annealing, European Journal of Operational Research 130 (1) (2001) 190-201.
[15] E. Elbeltagi, T. Hegazy, A hybrid AI-based system for site layout planning in construction, Computer-Aided Civil and Infrastructure Engineering, 16 (2) (2001) 79-93.
[16] K.W. Chau, M. Anson, A knowledge-based system for construction site level facilities layout, Lecture Notes in Artificial Intelligence, Springer-Verlag 2358 (2002) 393-402.
[17] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Publishing Co., Reading, Mass., 1989.
[18] K.W. Chau, Calibration of flow and water quality modeling using genetic algorithm, Lecture Notes in Artificial Intelligence, Springer-Verlag 2557 (2002) 720-720.
[19] K.W. Chau, F. Albermani, Genetic algorithms for design of liquid retaining structure, Lecture Notes in Artificial Intelligence, Springer-Verlag 2358 (2002) 119-128.
[20] K.W. Chau, F. Albermani, Knowledge-based system on optimum design of liquid retaining structures with genetic algorithms, Journal of Structural Engineering, ASCE 129 (10) (2003) 1312-1321.

[21] C.T. Cheng, C.P. Ou, K.W. Chau, Combining a fuzzy optimal model with a genetic algorithm to solve multiobjective rainfall-runoff model calibration, Journal of Hydrology 268 (1-4) (2002) 72-86.
[22] M.E. Dyer, L.A. Wolsey, Formulating the single-machine sequencing problem with release dates as a mixed integer-program, Discrete Applied Mathematics 26 (2-3) (1990) 255-270.

Table 1 Typical construction-oriented example of site-level facilities showing definitions of type number f and total number of facilities F

Site-level construction facility	**Type number f**
concrete batch plant	1
lay-down yard	2
receiving warehouse	3
reinforcement workshop	4
falsework workshop	5
…..	..
…..	..
…..	F

Table 2 Relation between opening, operation, and closure of transfer center j during period t

Phase	x_{jt}	y_{jt}	$z_{j,t-1}$	$z_{j,t}$
Initial	0	0	0	0
Opening	1	0	0	1
Operation	0	0	1	1
Closure	0	1	1	0

Table 3 Total quantities of source and demand of concrete resources (in cubic meters)

Source/ Demand	**Year**		
	1	**2**	**3**
S1	1200	1500	2000
S2	600	800	1000
S3	600	820	1060
D1	800	1000	1300
D2	700	900	1200
D3	400	520	700
D4	500	700	860

Table 4 Various costs for the concrete batch plants (in $)

Components	Transfer center	Year		
		1	2	3
Variable operation cost per unit per annum	**T1**	670	710	750
	T2	870	890	910
	T3	880	900	920
Fixed operation cost per annum	**T1**	133000	141000	149000
	T2	167000	170000	174000
	T3	166000	169000	173000
Opening cost	**T1**	667000	707000	749000
	T2	767000	782000	798000
	T3	766000	781000	797000
Closing cost	**T1**	333000	353000	374000
	T2	367000	374000	382000
	T3	368000	375000	383000

Table 5 Unit transportation costs amongst source, concrete batch plant, and destination (in $)

From	Year		
	1	2	3
S1 to T1	1470	1620	1780
S1 to T2	1670	1790	1910
S1 to T3	1690	1810	1930
S2 to T1	1330	1470	1610
S2 to T2	1730	1890	2060
S2 to T3	1750	1910	2080
S3 to T1	1070	1120	1180
S3 to T2	1600	1740	1900
S3 to T3	1630	1780	1940
S1 to D1	4000	4400	4840
S1 to D2	4130	4460	4820
S1 to D3	5000	5300	5620
S1 to D4	5330	5600	5880
S2 to D1	3670	4000	4360
S2 to D2	4070	4470	4880
S2 to D3	3870	4140	4430
S2 to D4	4800	5040	5290
S3 to D1	5670	6230	6860
S3 to D2	5200	5670	6180
S3 to D3	4330	4550	4780
S3 to D4	3870	3980	4100
T1 to D1	1270	1380	1630
T1 to D2	1470	1610	1900
T1 to D3	1570	1680	1930
T1 to D4	1400	1470	1680
T2 to D1	1100	1210	1330
T2 to D2	1200	1300	1400
T2 to D3	1230	1310	1390
T2 to D4	1470	1540	1620
T3 to D1	1080	1190	1310
T3 to D2	1180	1270	1380
T3 to D3	1210	1280	1360
T3 to D4	1440	1510	1590

Table 6 Detailed solution on concrete deliveries amongst sources, batch plants, and destinations for the example

From	Year		
	1	**2**	**3**
S1 to T1	1200	880	100
S1 to T2	0	620	1200
S1 to T3	0	0	700
S2 to T1	600	800	1000
S2 to T2	0	0	0
S2 to T3	0	0	0
S3 to T1	600	820	1060
S3 to T2	0	0	0
S3 to T3	0	0	0
T1 to D1	800	1000	1300
T1 to D2	700	800	0
T1 to D3	400	0	0
T1 to D4	500	700	860
T2 to D1	0	100	0
T2 to D2	0	520	500
T2 to D3	0	0	700
T2 to D4	0	0	0
T3 to D1	0	0	0
T3 to D2	0	0	700
T3 to D3	0	0	0
T3 to D4	0	0	0

Table 7 Example cases with a diversity of problem sizes

Test case	Problem size				
	I	**J**	**K**	**F**	**T**
1	3	3	4	1	3
2	10	8	10	5	5
3	30	20	30	20	10
4	40	30	40	50	10
5	60	50	60	75	10
6	80	60	80	100	10

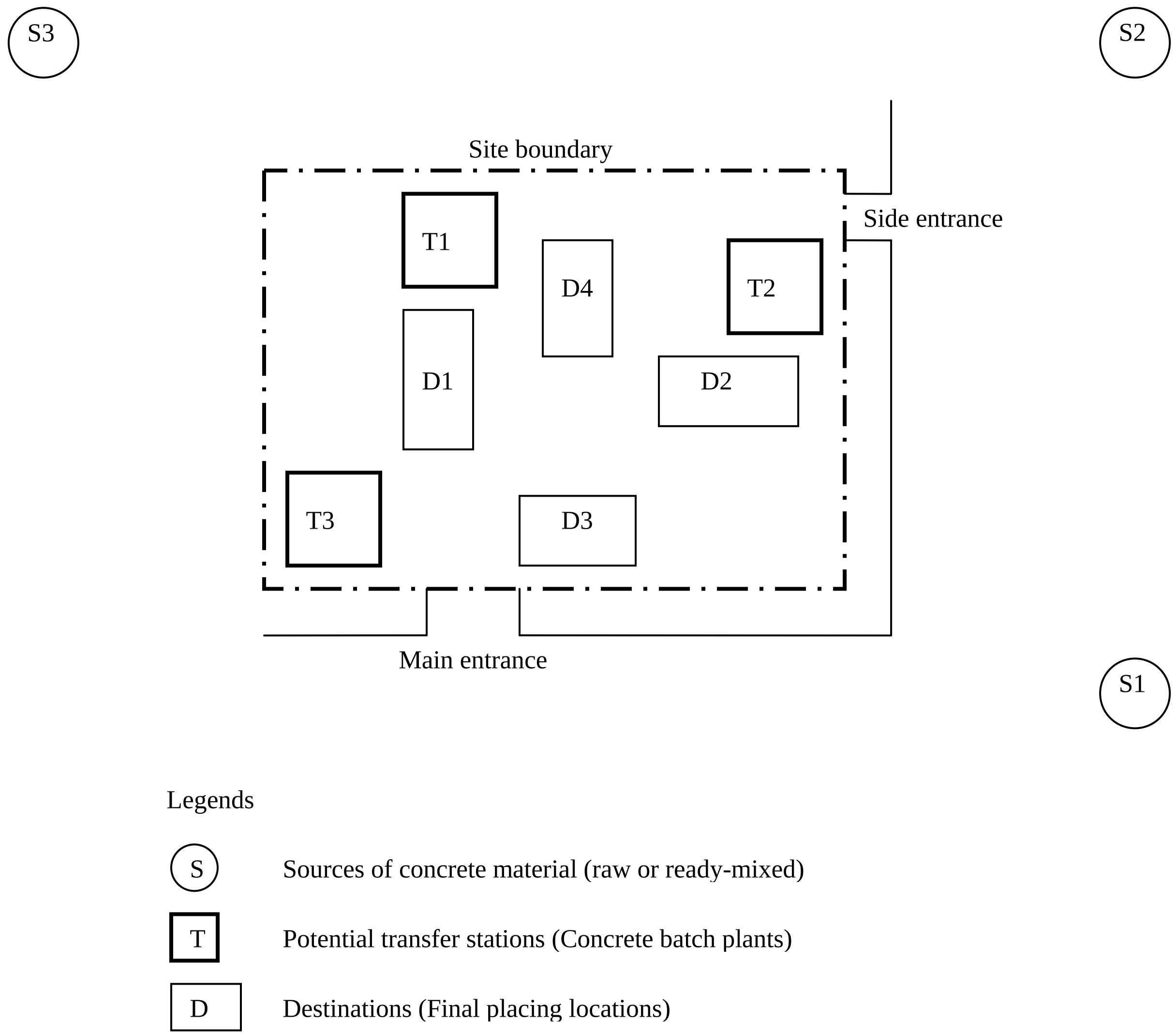

Figure 1 Schematic layout of the worksite for the example

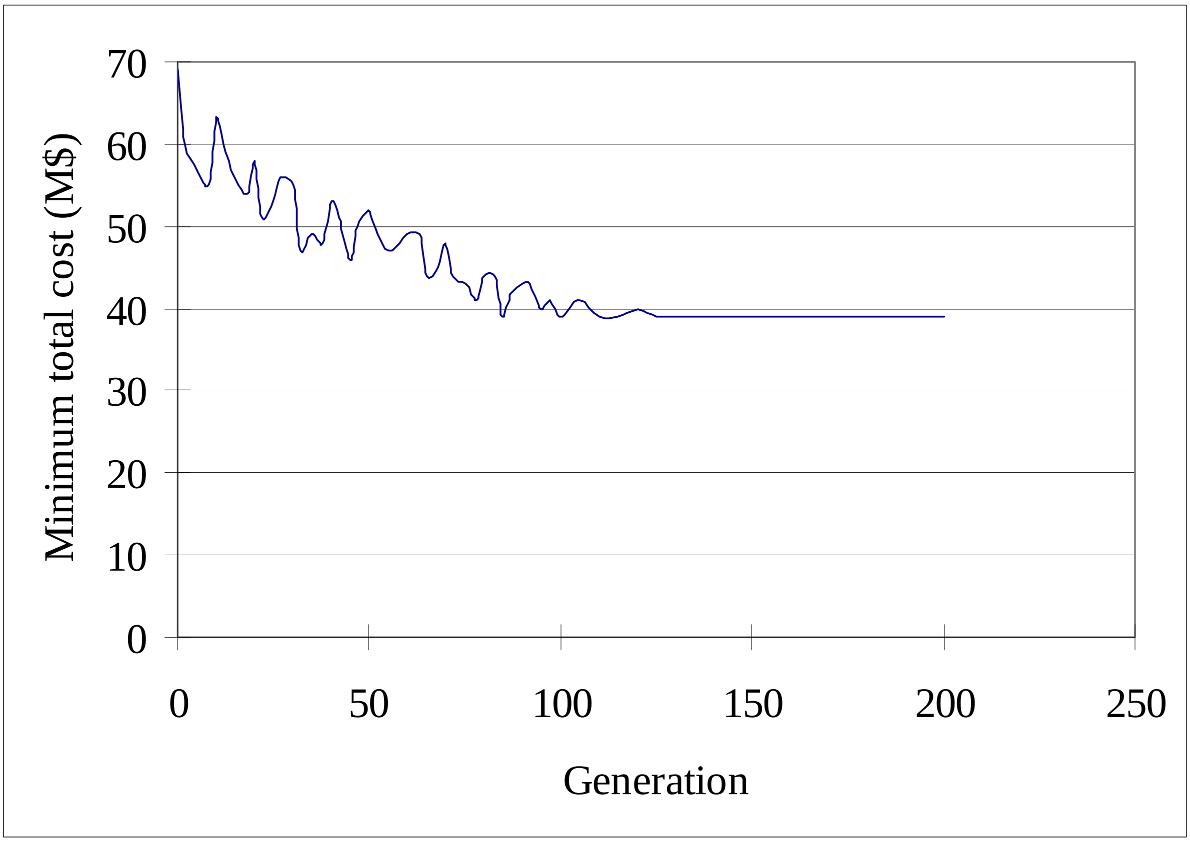

Figure 2 Solution by the split-step algorithm versus number of generation

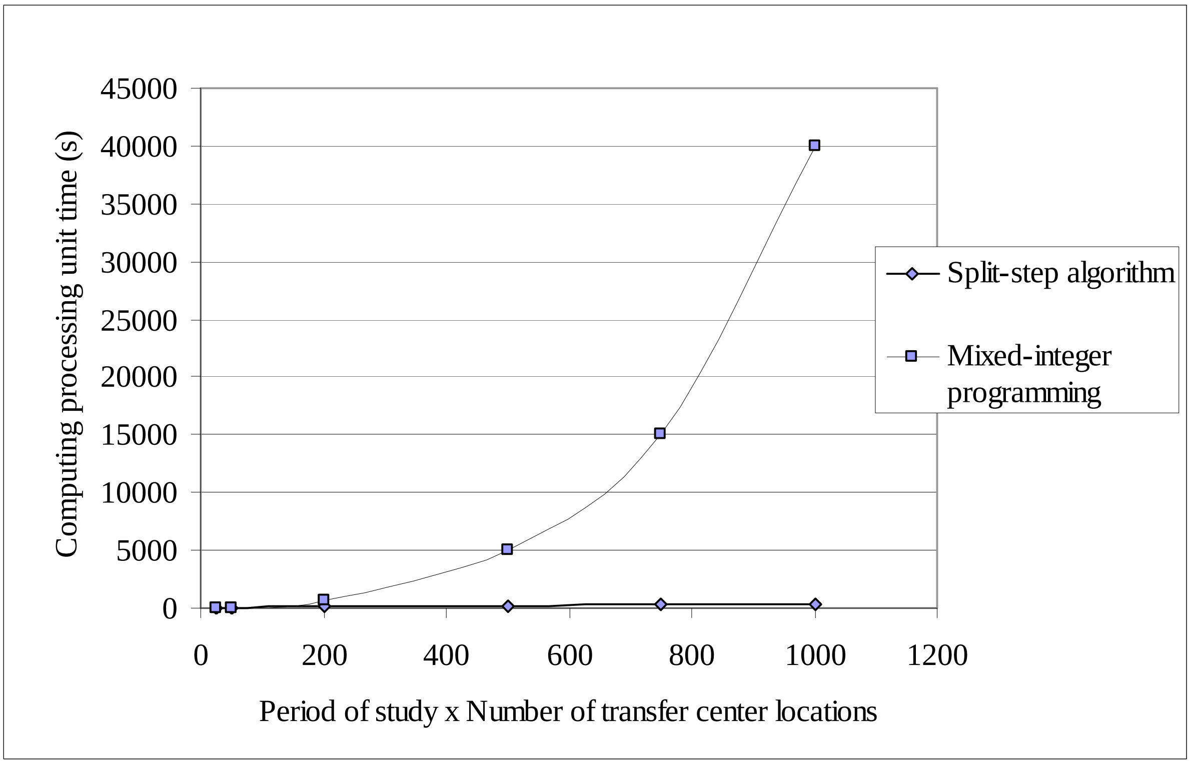

Figure 3 Relationships between the computing processing unit versus the multiplication of the period of study times the number of transfer center locations for both the mixed-integer programming and the split-step algorithm